IN THE CLAIMS:

This listing of the claims replaces all prior versions and listings of the claims. Please amend claims 6-8 and add claims 33-52 as follows:

Claim 1. (previously presented) A method for executing a network-based distributed application, the method comprising:

executing application instances of the distributed application by application containers, each application container sharing state information about its application instance with other application containers;

calculating quality of service metrics for each application instance by the application containers; and

distributing application workload among the application instances using a decentralized workload management layer based on the quality of service metrics.

Claim 2. (original) The method of claim 1, further comprising associating application containers with autonomous workload management elements, the workload management elements forming the workload management layer.

Claim 3. (original) The method of claim 2, further comprising coordinating the application instances through a coordination mechanism coupled to the workload management layer.

Claim 4. (original) The method of claim 1, wherein distributing application workload among the application

instances further comprises reducing workload assigned to an application container when the quality of service metrics reach an overload threshold value.

Claim 5. (previously presented) The method of claim 4, wherein reducing workload assigned to the application container further comprises:

examining an encoding of work unit groups provided by each application instance;

splitting a currently assigned work unit group into smaller work unit groups;

assigning at least one of the smaller work unit groups to other application containers; and

utilizing a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 6. (currently amended) The method of claim 1, wherein distributing application workload among the application instances further comprises increasing workload assigned to ~~the~~ an application container when the quality of service metrics reach an under-load threshold value.

Claim 7. (currently amended) The method of claim 6, wherein increasing workload assigned to the application container further comprises:

examining an encoding of work unit groups provided by each application instance;

combining at least two currently assigned work unit groups into a ~~smaller~~ larger work unit group;

assigning the ~~smaller~~ larger work unit group to the application container; and

utilizing a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 8. (currently amended) The method of claim 1, further comprising dividing workload assigned to a single application instance into ~~to~~ at least two application instances if a quality of service metric reaches an overload threshold.

Claim 9. (original) The method of claim 1, further comprising:

dividing a total workload performed by the distributed application among the application instances;

assigning each of the application instances a fractional workload; and

filtering client requests at the application containers based on the fractional workload assigned to the application instances.

Claim 10. (original) The method of claim 9, further comprising migrating a client from a first application container to a second application container if workload from the client is not assigned to the application instance executing at the first application container.

Claim 11. (original) The method of claim 10, further comprising labeling client requests such that application containers can determine if the requests belong to the fractional workload assigned to the application instances.

Claim 12. (original) The method of claim 1, further comprising receiving the application instances from application loaders.

Claims 13-32. (canceled)

Claim 33. (new) A system for executing a distributed computer application, the system comprising:

one or more application containers configured to execute an application instance of the distributed application, share state information about its application instance with other application containers and determine quality of service metrics for the application instance; and

one or more workload management elements forming a decentralized workload management layer, wherein each workload management element is configured to be associated to one of the application containers and to assign a workload to the application container based on the quality of service metrics received by the application container.

Claim 34. (new) The system of claim 33, wherein each workload management element is further configured to autonomously increase and decrease the assigned workload to its associated application container.

Claim 35. (new) The system of claim 34, wherein each workload management element is further configured to divide the assigned workload into two or more application containers if the assigned workload to its associated application container is to be decreased.

Claim 36. (new) The system of claim 34, wherein each workload management element is further configured to combine the assigned workload of two or more application containers if the assigned workload to its associated application container is to be increased.

Claim 37. (new) The system of claim 33, wherein each application container is further configured to pass inbound packets to executing application instances when the inbound packets belong to its assigned workload, and to pass inbound packets to its associated workload management element when the inbound packets do not belong to its assigned workload.

Claim 38. (new) The system of claim 33, further comprising workload tags coupled to data packets of application containers, the workload tags configured to allow application containers to identify whether the inbound packets belong to its assigned workload.

Claim 39. (new) The system of claim 33, further comprising a coordination mechanism configured to allow workload

management elements to locate each other and determine the current work assignments of each application container.

Claim 40. (new) The system of claim 33, further comprising an application loader configured to provide executable application code to application containers.

Claim 41. (new) A computer program product embodied in a tangible media comprising:

computer readable program codes coupled to the tangible media for executing a network-based distributed application, the computer readable program codes configured to cause the program to:

execute application instances of the distributed application in application containers, each application container sharing state information about its application instance with other application containers;

receive quality of service metrics for each application instance; and

distribute application workload among the application instances using a decentralized workload management layer based on the quality of service metrics.

Claim 42. (new) The computer program product of claim 41, further comprising program code configured to associate application containers with workload management elements, the workload management elements forming the workload management layer.

Claim 43. (new) The computer program product of claim 42, further comprising program code configured to coordinate the application instances through a coordination mechanism coupled to the workload management layer.

Claim 44. (new) The computer program product of claim 41, wherein the program code configured to cause the program to distribute application workload among the application instances further comprises program code to cause the program to reduce workload assigned to an application container when the quality of service metrics reach an overload threshold value.

Claim 45. (new) The computer program product of claim 43, wherein the program code configured to cause the program to reduce workload assigned to the application container further comprises program code to cause the program to:

examine an encoding of work unit groups provided by each application instance;

split a currently assigned work unit group into smaller work unit groups;

assign at least one of the smaller work unit groups to other application containers; and

utilize a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 46. (new) The computer program product of claim 41, wherein the program code configured to cause the program to

distribute application workload among the application instances further comprises program code to cause the program to increase workload assigned to an application container when the quality of service metrics reach an under-load threshold value.

Claim 47. (new) The computer program product of claim 46, wherein the program code configured to cause the program to increase workload assigned to the application container further comprises program code to cause the program to:

examine an encoding of work unit groups provided by each application instance;

combine at least two currently assigned work unit groups into a larger work unit group;

assign the larger work unit group to the application container; and

utilize a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 48. (new) The computer program product of claim 41, further comprising program code configured to divide workload assigned to a single application instance into at least two application instances if a quality of service metric reaches an overload threshold.

Claim 49. (new) The computer program product of claim 41, further comprising program code configured to:

divide a total workload performed by the distributed application among the application instances;

assign each of the application instances a fractional workload; and

filter client requests at the application containers based on the fractional workload assigned to the application instances.

Claim 50. (new) The computer program product of claim 49, further comprising program code configured to migrate a client from a first application container to a second application container if workload from the client is not assigned to the application instance executing at the first application container.

Claim 51. (new) The computer program product of claim 50, further comprising program code configured to label client requests such that application containers can determine if the requests belong to the fractional workload assigned to the application instances.

Claim 52. (new) The computer program product of claim 41, further comprising program code configured to receive the application instances from application loaders.